Week 12: Capstone Project Part 5.1_Salman_NUS

```
!pip install openai==0.28 requests python-dotenv
```

```
Collecting openai==0.28
  Downloading openai-0.28.0-py3-none-any.whl.metadata (13 kB)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (2.32.3)
Collecting python-dotenv
  Downloading python_dotenv-1.1.1-py3-none-any.whl.metadata (24 kB)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from openai==0.28) (4.67.1)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.11/dist-packages (from openai==0.28) (3.11.15)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests) (3.4.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests) (2.5.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests) (2025.7.14)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->openai==0.28) (2.6.1)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.11/dist-packages (from aiohttp->openai==0.28) (1.4.0)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->openai==0.28) (25.3.0)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.11/dist-packages (from aiohttp->openai==0.28) (1.7.0)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.11/dist-packages (from aiohttp->openai==0.28) (6.6.3)
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->openai==0.28) (0.3.2)
Requirement already satisfied: yarl<2.0,>=1.17.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->openai==0.28) (1.20.1)
Requirement already satisfied: typing-extensions>=4.2 in /usr/local/lib/python3.11/dist-packages (from aiosignal>=1.1.2->aiohttp->openai==0.28) (4.14.1)
Downloading openai-0.28.0-py3-none-any.whl (76 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 76.5/76.5 kB 2.7 MB/s eta 0:00:00
Downloading python_dotenv-1.1.1-py3-none-any.whl (20 kB)
Installing collected packages: python-dotenv, openai
  Attempting uninstall: openai
    Found existing installation: openai 1.97.0
    Uninstalling openai-1.97.0:
      Successfully uninstalled openai-1.97.0
Successfully installed openai-0.28.0 python-dotenv-1.1.1
```

```
WEATHER_API_KEY = "d0f2e4100b804758bce175542252107"
```

```
from transformers import pipeline

# Load a lightweight instruction-tuned model
extractor = pipeline("text2text-generation", model="google/flan-t5-base")
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
config.json:        1.40k/?  [00:00<00:00,  22.8kB/s]

model.safetensors:  100%                                        990M/990M  [00:18<00:00,  58.1MB/s]

generation_config.json:  100%                                   147/147  [00:00<00:00,  2.45kB/s]

tokenizer_config.json:        2.54k/?  [00:00<00:00,  65.8kB/s]

spiece.model:  100%                                             792k/792k  [00:00<00:00,  1.61MB/s]

tokenizer.json:        2.42M/?  [00:00<00:00,  14.6MB/s]

special_tokens_map.json:        2.20k/?  [00:00<00:00,  68.7kB/s]
Device set to use cpu
```

```
import requests

def get_weather(location):
    try:
        url = "http://api.weatherapi.com/v1/current.json"
        params = {
            "key": WEATHER_API_KEY,
            "q": location,
            "aqi": "no"
        }
        response = requests.get(url, params=params)
        response.raise_for_status()
        data = response.json()
        return f"Current weather in {data['location']['name']}: {data['current']['temp_c']}°C, {data['current']['condition']['text']}"
    except Exception as e:
        return f"Weather API Error: {str(e)}"
```

```
def extract_location_from_query(query):
    prompt = f"What city or location is this sentence referring to? \"{query}\""
    result = extractor(prompt, max_new_tokens=10)[0]["generated_text"]
    return result.strip()

def weather_agent(query):
    location = extract_location_from_query(query)
    print(f"[🔍] Detected Location: {location}")
    return get_weather(location)
```

```
test_queries = [
    "What's the weather like in Singapore?",
    "Is it raining in London right now?",
    "Tell me the temperature in New York",
    "Is it hot in Dubai?"
]

for query in test_queries:
    print(f"\n🧠 Query: {query}")
    print(f"🌦️ Response: {weather_agent(query)}")
```

```
🧠 Query: What's the weather like in Singapore?
[🔍] Detected Location: location
🌦️ Response: Current weather in Modjadje's Location: 15.1°C, Partly Cloudy

🧠 Query: Is it raining in London right now?
```

```
[🔍] Detected Location: London
🌦️ Response: Current weather in London: 22.3°C, Partly cloudy

🧠 Query: Tell me the temperature in New York
[🔍] Detected Location: New York
🌦️ Response: Current weather in New York: 24.4°C, Partly cloudy

🧠 Query: Is it hot in Dubai?
[🔍] Detected Location: Dubai
🌦️ Response: Current weather in Dubai: 37.1°C, Sunny
```